An Artificial Neural Network Based on Adaptive Resonance Theory for Fault Classification on an Automated Assembly Machine

Heshan Fernando and Brian Surgenor*

Department of Mechanical Engineering
Queen's University
Kingston, Ontario, K7L 3N6, Canada

Abstract

An unsupervised artificial neural network (ANN) based on the ART2-A algorithm is compared to a rule-based method for fault classification on an automated assembly machine. Machine data is collected using three greyscale sensors and two redundant limit switches for 11 different operating conditions. Descriptive features are extracted from the raw data and two data sets, each containing 180 feature vectors, are created for testing both methods. The first data set contains 'real' feature vectors obtained from the original sensor signals, and the second data set contains 'simulated' feature vectors obtained by scaling the 'real' feature vectors. The second data set is used to test the performance of each system when variations are present in the input space. During testing, the rule-based system correctly classified 98.3% of all feature vectors, but its classification thresholds needed to be manually adjusted to accommodate the 'simulated' data set. The ART2-A network perfectly classified the 'real' data set into 13 clusters, and then correctly classified the 'simulated' data into the same 13 clusters without any modification to the algorithm's tuning parameter, vigilance.

1. Introduction

A majority of automated machines are susceptive to developing faults (i.e. deviations from normal machine behaviour) [1]. In automated assembly machines, these faults can range from equipment faults, such as broken drive belts, seized actuators and damaged sensors, to task faults, such as ejected parts from feed chutes, part jams on transfer tracks and misaligned parts on carriers [2]. Undetected faults can lead to deteriorated product quality and production downtimes, so advanced supervisory systems are required for classifying different operating conditions on a machine. The two main tasks of a supervisory system are fault detection (i.e. identifying that an anomalous operation has occurred) and fault isolation (i.e. determining the type and location of a fault). To accomplish these tasks effectively, conventional supervisory systems monitor machine processes using multiple sensors, and employ rule-based change detection and classification methods for fault detection and fault isolation. Limit checking is an example of a commonly used change detection method, where faults are detected as a measured variable rises above or drops below a pre-defined threshold. In a supervisory system, limit checking can be applied to each sensor signal to detect different faults, and inference methods such as IF-THEN statements can be used to isolate faults to a particular type and/or location [1]. The drawbacks of such a system are:

1. Fault thresholds for each sensor need to be determined prior to designing the supervisory system.
2. High-quality sensors are required for reliable signals with high signal-to-noise ratios.
3. Processing logic must be manually programed for fault isolation.
4. Faults cannot be correctly classified if the features of input signals are not linearly separable for different operating conditions.

These drawbacks can incur large costs and long designs times when developing supervisory systems for complex machines, and they can also cause supervisory systems to fail if unexpected variations occur in the input signals. Alternatively, artificial intelligence (AI) methods such as neural networks (NN) have been proposed for developing advanced supervisory systems due to their powerful non-linear function approximation capabilities and adaptive learning abilities. These properties allow neural networks to automatically learn operating conditions of a machine without needing to develop a formal model. Such a system would allow quick development of supervisory systems for complex machines and could even be used with low-quality sensor signals as inputs. In this paper, a conventional rule

* Corresponding author: Tel: (613) 533-2568; Fax: (613) 533-6489; E-mail: surgenor@me.queensu.ca

based supervisory system and an advanced artificial neural network trained using an unsupervised Adaptive Resonance Theory (ART-2A) method are compared for classifying operating conditions on an automated assembly machine. Results are presented on each system's ability to classify a data set containing real measured values and a second data set containing simulated sensor values.

2. Neural Networks

The application of artificial neural networks (ANN) for fault detection and fault isolation has been widely published in recent decades. Tools for efficient training and implementation of various ANNs are readily available, and it is likely that more than half of the advanced supervisory systems today employ some form of ANN methods [1]. Their ability to provide decent classification results for complicated processes with relatively moderate design effort is highly desirable. Many researchers have developed ANN-based supervisory systems to detect and isolate faults in gears [3], bearings [4], motors [5], valves [6], pumps [7] and pneumatic systems [8]. Different types of ANNs with both supervised and unsupervised learning have been used. Supervised training using the Backpropagation (BP) algorithm on a multi-layer perceptron feed-forward (MLP-FF) network is the most widely used ANN learning method and is employed in many of the studies cited above. This method requires the network to first be trained using examples of input/output pairs of sample data prior to testing. The design of such a network is often a tedious task as certain parameters need to be carefully tuned. These parameters include: learning rate, number of hidden layers and number of hidden neurons. Furthermore, if the training data do not represent a good generalization of the input space, the network will not perform well during testing [8]. Alternatively, unsupervised neural networks, which automatically learn to classify similar types of data into clusters without any training, can be used if historical data is scarce or if a faster implementation is required. Studies by [7] and [8] have explored using unsupervised learning algorithms based on Adaptive Resonance Theory (ART) for fault detection and fault isolation of centrifugal pumps and pneumatic systems, respectively. The studies also compared the use of supervised Backpropagation learning methods on the same data and they both found that ART was much easier to implement and achieved the same level of accuracy for classifying operating conditions in each study. Therefore, in the current study, an ART algorithm is chosen for developing a supervisory system for classifying operating conditions on an automated assembly machine.

2.1. Adaptive Resonance Theory

The ART algorithm, developed by [9] is an unsupervised neural network learning algorithm that has demonstrated to be a highly effective classifier. It is unique compared to other unsupervised networks as the total number of clusters, also known as prototypes, does not need to be known prior to execution. ART can automatically create a prototype to represent new classes of input data, or classify an input into an existing cluster if it deems the input and the prototype to be *sufficiently* similar. The idea of sufficiency of similarity is given as a vigilance parameter, ρ, which can vary between 0 and 1. High vigilance values result in more clusters of closely related data, and low vigilance values result in fewer clusters of more generally related data. The user must carefully tune this parameter to obtain the desired amount of clusters. There have been many variations of ART since its inception. ART1 [9], which was used by [7], accepts only binary input data. ART2 which was used by [8] accepts continuous values as input data. ART2-A [10], which is used in the current study, is an enhanced version of the ART2 algorithm with improved execution time.

3. Experimental Setup

Fault detection and fault isolation were tested on an automated assembly machine that assembles parts onto continuously moving carriers. Sensors that could detect the presence of parts and carriers were mounted at different locations on the machine for monitoring the assembly process. Data from these sensors were collected for the machine operating under different conditions: normal operation, 6 task faults and 4 sensor faults. The data was then used to test each supervisory system's ability to classify the normal operation of the machine and the 10 different fault conditions. In this section, the machine's operation, fault conditions, and sensors are described.

3.1. Automated Assembly Machine

A slightly modified version of an industrial automated assembly machine, as shown in Figure 1 was used as a testbed for obtaining operation data. The machine assembles black-coloured, 8 mm, round *parts* onto white-coloured

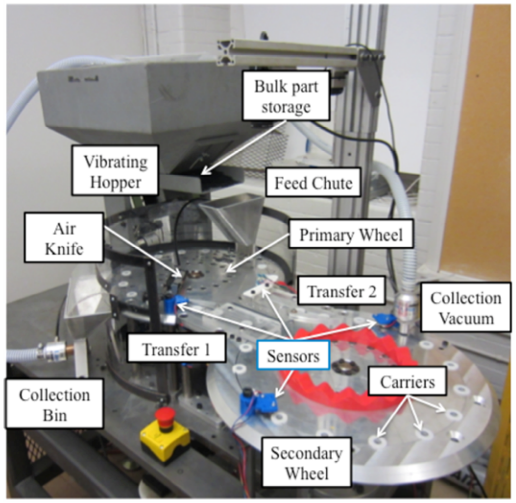

Figure 1. Key features of the automated assembly machine used for testing.

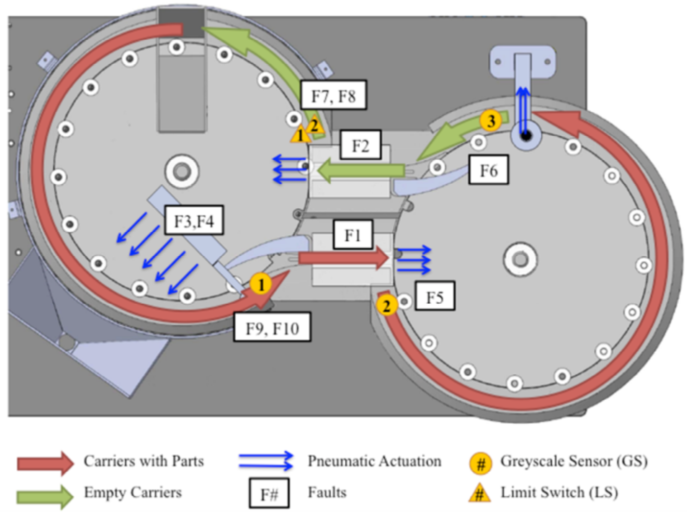

Figure 2. Top-down schematic of the automated assembly machine showing normal operation conditions and sensor placement.

continuously moving *carriers* at a rate of 108 assemblies per minute. Parts are stored in a bulk supply inside a *vibrating hopper*. During normal operation, a steady stream of parts falls through the *feed chute* on to the carriers below. There, a single part is assembled onto a groove on the carrier. The carriers then move around the *primary wheel* and the excess parts are blown into the *collection bin* by an *air-knife*. Next, the part/carrier assemblies are transferred to the *secondary wheel* through an air-track at *transfer 1*. The assemblies travel around the secondary wheel until they reach the *collection vacuum*, where the part is removed from the carrier. The empty carriers then return to the primary wheel through another air-track at *transfer 2* and the assembly process is repeated.

3.2. FAULTS

Pneumatics are primarily used for actuation on the automated assembly machine, so variation in supply pressure can cause different faults to occur. For example, a decrease in supply pressure to the transfer track nozzles can induce part jams in the track, while an increase in supply pressure to the air knife can cause the assembled parts and carriers to blow off the primary wheel. Figure 2 shows a schematic of a top-down view of the machine. The normal operation of the machine is shown with coloured arrows: red arrows indicate regions along the assembly path where a part should be present on the carrier, and green arrows indicate regions where the carriers should be empty. Common faults and their locations are labeled F1 to F10 on the schematic, and they are described in Table 1. Faults F1 to F6 are task faults

Table 1. Descriptions of faults and their causes on the automated assembly machine.

Fault	Description	Causes
F1	Parts and carriers jammed at first transfer track	Low air pressure to track's nozzle, or loosely fitted part ejected and lodged in track
F2	Carriers jammed at second transfer track	Low air pressure to track's nozzle
F3	Part missing after air knife	Increase in air pressure to air knife, or loosely assembled part on carrier
F4	Carrier missing after air knife	Increase in air pressure to air knife
F5	Part missing after Transfer 1	Part ejected from carrier at the first transfer track
F6	Part present after collection vacuum	Low air pressure to vaccum
F7	LS1 signal continuously LOW	Loose or broken wire connection on LS1
F8	LS1 signal continuously HIGH	Broken switch arm on LS1
F9	GS1 signal continuously HIGH	Broken LED on GS1 sensor
F10	GS1 signal continuously LOW	Loose or broken wire connection on GS1

in the assembly process, and faults F7 to F10 are equipment faults related to sensor failure. The six task faults are simulated by manually controlling the air supply to the various pneumatic actuation components on the machine. Sensor faults are simulated by manually loosening wire connections on the sensors themselves.

3.3. Sensors

Three greyscale sensors (GS1, GS2 and GS3) and two redundant limit switches (LS1 and LS2) are used together as a sensor system to monitor the operation of the assembly machine testbed. Sensor placement is shown on Figure 2: GS1 is located at the entrance of *transfer 1* to confirm that both carriers and parts are present after the air-knife; GS2 is located at the exit of *transfer 1* to confirm that both carriers and parts are entering the *secondary wheel*; GS3 is located after the *collection vacuum* to confirm that the part is removed from the carrier; and two limit switches (LS1 and LS2) are located at the exit of *transfer 2* to confirm that empty carriers are entering the *primary wheel*. The sensors are placed at locations that allow the earliest possible detection of the six task faults without interfering the machine's operation. Limit switches are used after transfer 2 instead of another greyscale sensor to have some variety in the data. Greyscale sensors use an LED to illuminate a surface and a photoresistor to measure the intensity of the reflected light. The photoresistor's resistance decreases with increasing intensity of light, so the sensor's output voltage is lower for a whiter surface than for a darker surface. As the carriers are white and the parts are black, the greyscale sensor can detect the presence of these components throughout the assembly process. Two limit switches are used to detect empty carriers after the second transfer. They are wired as 'normally closed', meaning that they output HIGH when there are no carriers and output LOW when a carrier passes through. These switches were mounted together for redundancy. Unlike the greyscale sensors, the limit switches are contact-sensors so they are prone to wear and tear. If the limit switch were to fail 'normally closed' (i.e. the switch arm breaks off), then the output would continuously read HIGH. This falsely indicates a fault F2 (Transfer 2 Jam). When two limit switches are used redundantly, the false positive for F2 is avoided, as the second limit switch continues to operate normally.

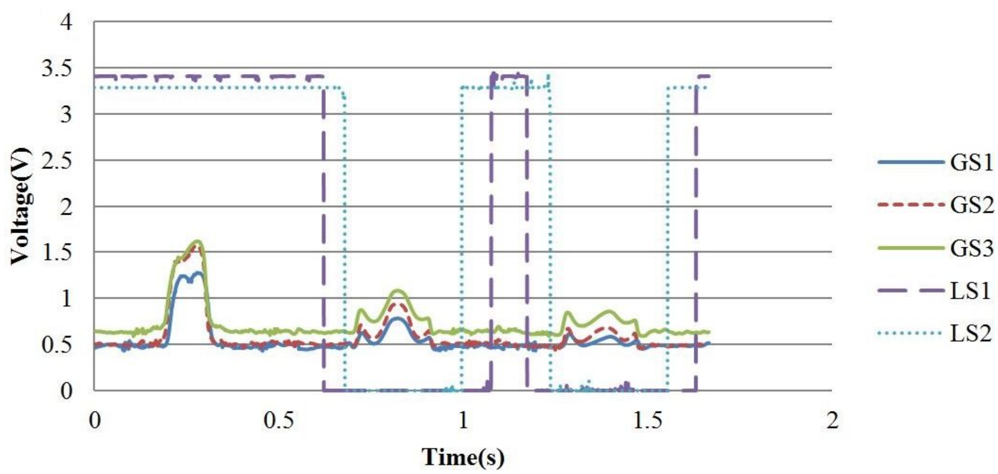

Figure 3. Output signals of each sensor for a missing carrier (0 s to 0.556 s), carrier with part (0.557 s to 1.112 s) and empty carrier (1.113 s to 1.668 s) at normal machine operation speed (108 assemblies per minute).

The analog voltage measurements from each sensor were collected using a data acquisition system at a sampling rate of 1000 Hz. Figure 3 shows the output signals for a 'missing carrier', 'carrier with part' and 'empty carrier' as they pass beneath each sensor, in sequence, at normal assembly speed. The greyscale scale sensors have a unique output for each condition, but some variation can be observed between the three sensors. This is likely due to the use of a very basic and low-cost photoresistor and variation in the overhead lighting at different areas of the automated assembly machine. The difference between 'carrier with part' and 'empty carrier' signals is subtle, but the greyscale sensors are still able to identify the presence of the small black part on the white carrier. The 'missing carrier' is highly distinguishable from the other two conditions as there is a large variation in output voltage for this condition. The limit switches are not able to detect the presence of the part, but they can distinguish between the 'missing carrier' and 'empty carrier'. This is acceptable as only empty carriers are expected after the second transfer (see Figure 2); fault F6, where a part is present after the collection vacuum, should be detected by GS3. However, there are some differences between the two limit switch signals. Despite being mounted together and measuring the same variable, the limit switches appear to go LOW for different amounts of time for the presence of the same carrier. This is due to the round shape of the carrier. LS1 is mounted such that its arm contacts the middle of the carrier and LS2 is mounted such that its arm contacts the edge of the carrier. Therefore, LS2 is in contact with the carrier for a shorter period of time compared to LS1.

4. METHODOLOGY

An unsupervised artificial neural network (ANN) based method and a rule-based thresholding method were compared for detecting and isolating faults on the automated assembly machine. First, sensor data were collected for the machine operating under normal condition as well as the ten fault conditions listed in Table 1. Task faults and sensor faults were simulated directly on the machine so that real sensor data could be obtained for all operating conditions. Next, descriptive features from the raw signal data were extracted using pre-processing techniques. These features included the maximum amplitude of each greyscale sensor and the length of time that the each limit switch read LOW, in a single assembly period. The features were then stored in 5-element feature vectors and a testing data set was created using multiple feature vectors from each operating condition. A second data set with simulated feature vectors was also created to test each method's performance when slight variations in the data were present. Finally, an unsupervised ANN trained using Adaptive Resonance Theory (ART-2A) and a custom rule-based thresholding method were used to classify the same sets of data. The performance and ease of implementation of each method were evaluated.

4.1. FEATURE EXTRACTION

Figure 3 showed raw signal data taken from the five sensors on the automated assembly machine for three assembly periods. Upon close inspection of the signals, one can see that the raw data cannot be used as-is for classifying different operating conditions. Firstly, the sensors are sampled at a rate of 1000 Hz which results in extremely large segments of data. The signals shown in Figure 3 represent only 1.668 s of machine operation, but contain over 1.7 million points of data in total. It would be impractical to design an ANN or a rule-based thresholding method to directly process this amount of data. Secondly, there is some overlap in the sensor signals for different carrier states, so thresholding is not possible on the raw data. For example, each greyscale sensor's 'empty carrier' signal lies within the upper and lower limits of the 'carrier with part' signal, so the two signals cannot be easily differentiated. In order to obtain good results for classifying operating conditions with both test methods, signals must be pre-processed to extract the most descriptive features for classifying operating conditions. For the automated assembly machine, the highest amplitude from each greyscale sensor and the length of time that each limit switch read LOW, per assembly period, were selected as five features for inputs. From Figure 3, it is clear that these features are the most descriptive for classifying operating conditions on the automated assembly machine. They are also easily extracted from the sensor signals.

The five features were extracted by first dividing segments of raw data into lengths equal to an assembly period (i.e. 0.556 s). Then from each assembly period the highest amplitude was found for each greyscale sensor using a simple *max()* operation in MATLAB®, and the lengths of time that each limit switch read LOW were found by counting the number data points that had a value of less than 0.1 V (to account for noise) and dividing that number by 1000 to give the length of time in seconds. The features were then stored in a 5-element feature vector and labelled appropriately with the type of operation that each vector represented. A test data set was created with multiple feature vectors for each operating condition.

4.2. TEST DATA SETS

Two data sets were created to test the performance of the ART-2A based ANN method and the rule-based thresholding method for fault detection and fault isolation on the automated assembly machine. The first data set consisted of 180 feature vectors extracted from the raw signal data: 30 feature vectors that represented normal machine operation and 15 feature vectors for each fault condition. More feature vectors were used for normal operation because that is the most common operating mode of the machine. After the first data set was created, it was found that the extracted feature vectors were very consistent for each operating condition, with very little within-class variation. In fact, the average within-class variation was calculated to be 2.26% for all 11 operating conditions. To test the robustness of both methods, a second 'simulated' data set was created by modifying the values of the 'real' data set. This was done by scaling a feature vector in the 'real' data set with a value between -7% and 7% (in 1% increments). To be consistent, pairs of feature vectors from normal operation class and individual feature vectors from a fault class were scaled by a *unique* scaling factor. This resulted in 180 'simulated' feature vectors for the second data set giving a total of 360 feature vectors for testing the two classification methods.

4.3. RULE-BASED METHOD

Rule-based methods typically involve limit-checking sensor signals for symptom generation and using inference methods such as IF-THEN statements to classify symptoms into machine operation conditions. Developing such methods require knowledge of sensor values for symptom generation, and knowledge of machine processes for classifying operating conditions. For the automated assembly machine, limit-checking methods were applied to extracted feature data to generate symptoms, and the combinations of symptoms, as summarized in Table 2, were used to classify operating conditions. This first required determining the upper and lower limits of each sensor's feature values for different symptoms. These limits were initially found by analyzing the feature vectors in the 'real' data set, and then calculating the maximum and minimum values of each feature for every operating condition. Next, these limits and the symptom chart shown in Table 2 were programmed using a series of IF-THEN statements to develop a rule-based fault detection and fault isolation system for the automated assembly machine. As the system was to be tested using both 'real' and 'simulated' data sets, the best performance was found by running 15 tests where the original upper and lower limits for symptom generation were increased and decreased, respectively, by 1% for each test.

Table 2. Symptoms for different operating conditions on the machine.

Operation Condition	GS1	GS2	GS3	LS1	LS2
Normal	+	+	O	O	O
F1	+	X	O	O	O
F2	+	+	O	X	X
F3	O	+	O	O	O
F4	X	+	O	O	O
F5	+	O	O	O	O
F6	+	+	+	O	O
F7	+	+	O	L	O
F8	+	+	O	H	O
F9	H	+	O	O	O
F10	L	+	O	O	O

Symptoms	
+	Carrier with Part
O	Empty Carrier
X	Missing Carrier
L	Low Voltage
H	High Voltage

4.4. ADAPTIVE RESONANCE THEORY (ART-2A) METHOD

The ART-2A algorithm for classifying operating conditions on the automated assembly machine was implemented using a software package by [11] released under the general public license (GPL). The program takes a data file containing labelled input feature vectors and applies the ART-2A algorithm to classify the data into clusters of similar data. The vigilance value is set by the user prior to execution. This parameter, which ranges from 0 to 1, controls the threshold at which the algorithm creates a new cluster. Ideally, a correctly tuned ART-2A algorithm would create 11 clusters to classify the 11 operating conditions on the automated assembly machine; however, depending on how the data is mapped in the input space, this exact number of perfect clusters may not be possible, and the ART-2A algorithm may create extra clusters to correctly classify the data. This is acceptable as long as each cluster contains data from only a single class. If multiple classes of data are present in a single cluster, then it is considered a misclassification. Once execution is complete, the program saves the labeled feature vectors in each cluster into output files for inspection by the user. This way, the exact contents of each cluster can be determined to ensure that proper classification is achieved. The algorithm was tested using the two data sets for varying vigilance values. The 'real' data set was first presented to the network and the vigilance value that resulted in the best classification was determined. The 'simulated' data was then presented to the network at the same vigilance value to determine whether the network classified the 'simulated' data into the same clusters.

5. Results and Discussion

When the rule-based method was first tested, the upper and lower limits for symptom generation were obtained by analyzing the 'real' data set and finding the maximum and minimum values for each type of feature. Of course, this guarantees a perfect classification for values in the 'real' data set, but not for the values in the 'simulated' data set, which were scaled by as much as ± 7%. For better classification of the 'simulated' data the upper and lower limits for symptom generation had to be increased and decreased, respectively, and this was done in 1% increments to find the optimal performance of the system. The results are shown in Figure 4. As expected, the overall performance of the system starts out very low, at 56.9%, with no scaling applied to the limits. This is because the performance on the simulated data is very poor. When the limits are scaled by a factor of ± 7%, it would be assumed that perfect classification should be achieved, but this is not the case. The maximum classification performance for the rule-based system is 98.3% which remains constant for limits scaled by ± 7% to ± 9%. At a ± 10% scaling factor, the overall performance begins to drop. The cause of the imperfect classification was found to be due to features extracted from GS1 sensor's signal. For a 'carrier with part' and 'empty carrier' the GS1 feature values were different by only 9% in some cases. This caused 5 simulated 'normal' feature vectors to be classified as fault F3, and one simulated F3 vector to be classified as 'normal'.

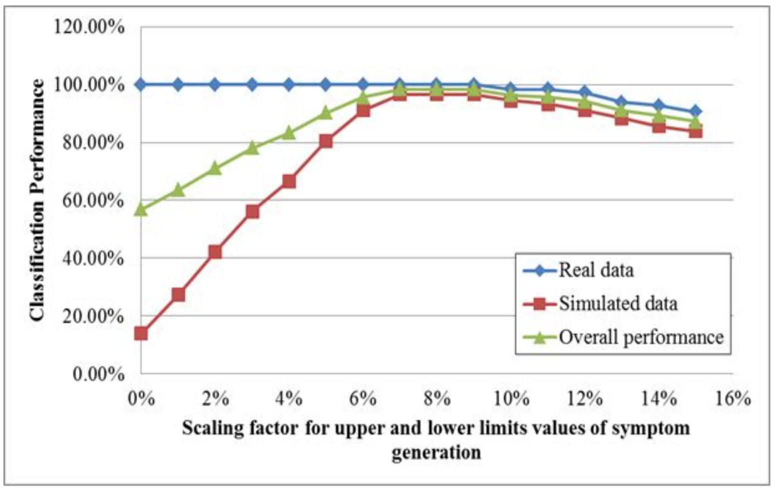

Figure 4. A plot showing the performance of the rule-based system as the upper and lower limits for symptom generation are increased and decreased, respectively, by a scaling factor.

The ART2-A based supervisory system was first tested by inspecting the clusters created for varying vigilance values for only 'real' data and then for both 'real' and 'simulated' data. This was done to ensure that the network created the same amount of clusters for the same vigilance values. The results are shown in Figure 5. Eleven clusters are formed at vigilance values of 0.9977 to 0.9986. However, inspection of these clusters revealed that in all cases, one cluster that contained all F7 feature vectors also contained a single misclassified 'normal' feature vector. Perfect classification was achieved at a vigilance value of 0.9987, but 13 clusters were created to classify the 11 classes of data; F8 and 'normal' feature vectors were both classified under two clusters per class. An important thing to note in the results is that perfect classification was achieved at the same vigilance value for both test cases. This means that even though the network was only 'trained' using the 'real' data, it still perfectly classified the 'simulated' feature vectors under the same clusters.

6. Summary and Future Work

Fault detection and Fault isolation on an automated assembly machine were tested using a rule-based method and an unsupervised artificial neural network based method. Both methods were required to classify 11operation conditions on the machine including normal operation, six task fault occurrences and four sensor fault occurrences. Three greyscale sensors and two redundant limit switches were employed to monitor the entire assembly process that involved continuous assembly of black-coloured parts onto white-coloured carriers. Sensor data were collected for each operating condition. This data was then pre-processed and descriptive features were extracted to present to each system as inputs. Two data sets containing multiple feature vectors for each operating condition were created for testing each classification system. The first data set contained only feature vectors obtained from 'real' sensor data, and the second data set contained feature vectors that were 'simulated' by scaling feature vectors in the first data set. A

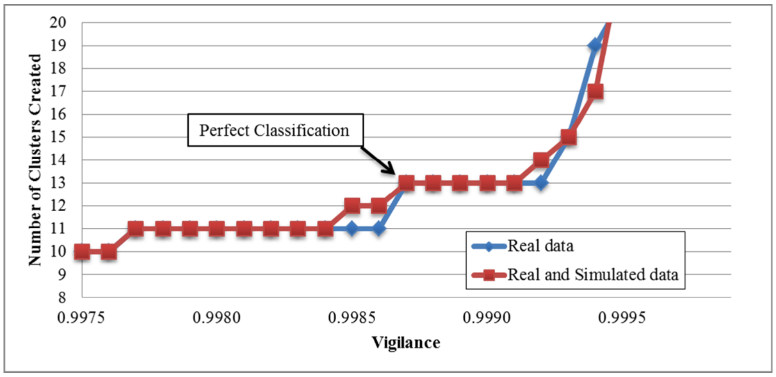

Figure 5. A plot showing the number of clusters created for varying vigilance values with both 'real' and 'simulated' data.

rule based system was developed for classifying feature vectors based on limit checking for symptom generation and inference methods for fault classification. While the system was able to perfectly classify the 'real' data set, it achieved only a maximum classification performance of 98.3% when the simulated data set was included. It was found that this was due overlapping of certain features that were not linearly separable by limit checking alone. The unsupervised ANN network was implemented using an ART2-A algorithm, which automatically classifies input data into clusters of similar data based on a user-tuned *vigilance* parameter. During testing it was found that the algorithm created 11 clusters when the vigilance was set between 0.9977 and 0.9986 for the 'real' data set; however, inspection of the clusters revealed that one cluster had a misclassified feature. Perfect classification was achieved at a vigilance value of 0.9987, but 13 clusters were created. When the testing was repeated for the 'simulated' data set, the algorithm correctly sorted the data into the same 13 clusters at the same vigilance value for a perfect classification. Work continues on the automated assembly machine for unknown and multiple fault detection techniques. Future research will also involve application of machine vision with neuro-fuzzy algorithms for advanced fault detection and isolation methods.

REFERENCES

[1] R. Isermann, Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance, Springer, 2006.

[2] G. Szkilnyk, K. Hughes, H. Fernando and B. Surgenor, "Spatiotemporal volume video event detection for fault monitoring in assembly automation", Proceedings of the 19th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Auckland, New Zealand, 2012.

[3] B. A. Paya, I. I. Esat and M. N. M. Badi, "Artificial neural network based fault diagnostics of rotating machinery using wavelet transforms as a pre-processor", Mechanical Systems and Signal Processing, Vol. 11, No. 5, pp. 751–765, 1997.

[4] B. Samanta and K. R. Al-Balushi, "Artificial neural network based fault diagnostics of rolling element bearings using time-domain features", Mechanical Systems and Signal Processing, Vol. 17, No. 2, pp. 317–328. 2003.

[5] X. Liu, H. Zhang, J. Liu and J. Yang, "Fault detection and diagnosis of permanent-magnet DC motor based on parameter estimation and neural network", IEEE Transactions On Industrial Electronics, Vol. 47, No. 5, pp. 1021–1030. 2000.

[6] P. Subbaraj and P. Kannapiran, "Artificial neural network approach for fault detection in pneumatic valve in cooler water spray system", International Journal of Computer Applications Vol. 9, No. 7, pp. 43–52. 2010.

[7] S. Rajakarunakaran, P. Venkumar, D. Devaraj and K. S. P. Rao, "Artificial neural network approach for fault detection in rotary system", Applications in Soft Computing, Vol. 8, pp. 740–748, 2008.

[8] M. Demetgul, I. N. Tansel and S. Taskin, "Fault diagnosis of pneumatic systems with artificial neural network algorithms", Expert Systems and Application, Vol. 36, pp. 10512–10519, 2009.

[9] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance", Cognitive Science Vol. 11, No. 1, pp. 23–63. 1987.

[10] G. A. Carpenter, S. Grossberg and D. B. Rosen, "ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition", Neural Networks Vol. 4, No. 4, pp. 493–504, 1991.

[11] T. Hudik and J. Zizka, "Adaptive Resonance Theory", http://www.ica.luz.ve/~dfinol/NeuroCienciaCognitiva/Sofware/art.htm (accessed December 1, 2013).